

WIRELESS DISTRIBUTED DATA SYNCHRONIZATION

Lauren Englisbe, Computer Science Department, Siena College, Loudonville, 12211, lauren.englisbe@students.siena.edu
Christopher Waggoner, Computer Science Department, DePauw University, Greencastle, IN 46135, cwaggoner@depauw.edu
Carl Singer (*), Computer Science Department, DePauw University, Greencastle, IN 46135, singer@depauw.edu

Handheld computers have become pervasive tools in the past few years. Their compact size and ease of mobility allow them to be useful in many different situations, not simply as pocket day planners. With the addition of a wireless Internet connection, the user is capable of communicating with other handhelds, traditional computers, and networks. This opens the door for data sharing among handhelds, other mobile computers, and servers, wherein each device may create and manipulate shared data, and synchronize it across all involved devices.

To explore issues associated with database synchronization between handheld computers and a central database server, we developed a prototype application called *PocketQuiz*. Our system allows teachers to create exams that can be downloaded to handheld computers by all the students in a class. Students can take exams offline; then when they later connect to the Internet completed exams can be submitted to the central server for grading.

A distributed database is at the center of the application. Faculty members use a web-interface to create and maintain exams. Students must be online and authenticated before being allowed to download exams. Upon authentication, a student chooses an exam and downloads it as an exam instance to a handheld computer. Once the exam is replicated, the student may take the exam at any time and in any place regardless of whether or not they are connected to the Internet. Exam answers are stored in the local handheld database until they are submitted to the server via an Internet connection. Upon submission, the central database is updated, and a database trigger is used to grade the exam. To store and manipulate the data on the handheld, we use SQL Server CE in the Windows CE environment. SQL Server running under Windows 2000 Server houses the central database. The application was developed using Microsoft Visual Basic .NET and tested on iPAQ Pocket PC's.

Some form of data replication is required to synchronize an exam and its answers between the server and a handheld computer. PocketQuiz uses Microsoft's Remote Data Access (RDA), which is a simple tool with functions the client uses to initiate data exchanges: push (update a server table that has been previously pulled), pull (create a table on the handheld computer by downloading a subset of a table on the server), and submitSQL (execute on the server a non-select SQL statement).

When PocketQuiz requests a copy of an exam, an *exam instance* is dynamically created. The exam instance is identified by a primary key consisting of the ID of the student requesting the copy, the ID of the class for which the exam is being requested and the exam ID itself. Once created, an exam instance will reside on both the handheld computer and the server, resulting in only two copies of the instance existing at any given time. Thus, the exam instance table is one that is partitioned by student. Furthermore, when a student takes an exam, the answers are recorded in an answer table on the handheld computer and uploaded to the server, which results in another table that is partitioned by student.

One issue we studied is the effect on database state from a loss in connectivity. For example, if a student requests a copy of an exam instance and there is a break in the connection during transmission, the exam instance might never be received by the handheld computer. Yet the server might assume it has been downloaded, and will reflect that state in the server database. Since we allow a copy of an exam instance on only one handheld computer, the student is no longer able to obtain a copy. We address these kinds of issues by simulating distributed transaction processing (commit and rollback) at the application level.

Since a student can complete an exam on a handheld computer and submit it later and since most exams must be completed by a certain time, we also considered how we might determine if the deadline was met, given that the handheld time cannot be trusted. We considered both hardware and software solutions.

In the future we plan to further explore time-based issues, security issues such as the 'lost handheld' problem, and add to PocketQuiz functionality and user friendliness.

This work was supported by National Science Foundation Grant Numbers EIA-0242293 and EIA-0215834